Web Authorization Protocol
Internet-Draft
Intended status: Standards Track
Expires: 5 December 2026

Y. Zehavi
Raiffeisen Bank International

3 June 2026

OAuth 2.0 RAR Metadata and Error Signaling
draft-zehavi-oauth-rar-metadata-03

Abstract

OAuth 2.0 Rich Authorization Requests (RAR) [RFC9396], standardizes the exchange and processing of authorization details but does not define metadata to describe authorization details types.

The document addresses a practical interoperability challenge regarding metadata of authorization details types, allowing clients tp dynamically discover metadata instead of relying on out-of-band agreements. It also standardizes error signaling, in case insufficient RAR was provided and offers structured ways of remediation.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://yaron-zehavi.github.io/oauth-rich-authorization-requests-metadata/draft-zehavi-oauth-rar-metadata.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-zehavi-oauth-rar-metadata/.

Discussion of this document takes place on the Web Authorization Protocol Working Group mailing list (mailto:oauth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/oauth/. Subscribe at https://www.ietf.org/mailman/listinfo/oauth/.

Source for this draft and an issue tracker can be found at https://github.com/yaron-zehavi/oauth-rich-authorization-requests-metadata.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction

OAuth 2.0 Rich Authorization Requests (RAR) [RFC9396] allows OAuth clients to request detailed and structured authorization, enabling advanced authorization models across domains such as banking and healthcare.

However, RAR [RFC9396] does not specify how clients discover metadata describing valid authorization details objects. Such metadata and documentation are obtained out-of-band.

This document defines:

* A new authorization server endpoint: authorization_details_types_metadata_endpoint, providing metadata for authorization details types, including documentation and JSON Schema definitions [JSON.Schema].

* Adds *required* authorization details types to OAuth 2.0 Protected Resource Metadata [RFC9728] response.

* A new normative OAuth 2.0 WWW-Authenticate Error Code, for resource servers to indicate insufficient_authorization_details as the cause of error.

* An OPTIONAL response body that MAY accompany the insufficient_authorization_details error, providing an informative and actionable authorization details object. This object can be used directly in a follow-up OAuth request.

* RECOMMENDED handling of large RAR [RFC9396] authorization details objects when issuing JWT access tokens, to avoid failures due to token sizes exceeding header size restrictions.

The optional providing of actionable authorization details objects by resource servers enables:

* Simplification for clients who can directly remediate without learning to construct valid authorization details objects.

* Support for ephemeral, interaction-specific claims from resource server, such as for example a risk profile or an internal interaction identifier, guiding authorization servers on required authentication strength and consent flows.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

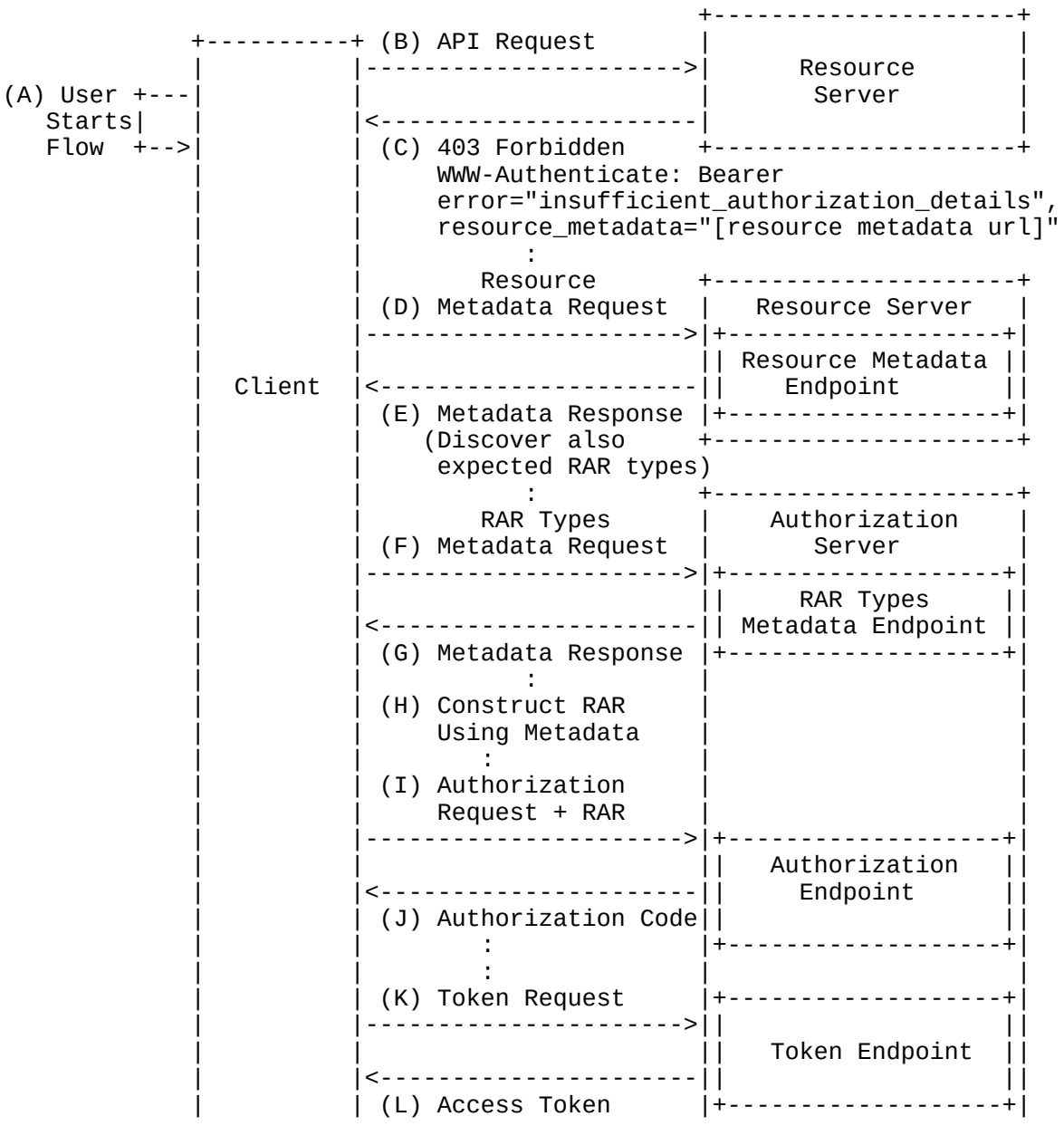
3. Protocol Overview

There are two main proposed flows:

* Client remediates using *metadata of required authorization details types*.

   * Client remediates using *actionable authorization details objects* provided by resource server.

3.1. Client remediates using metadata of required authorization details types



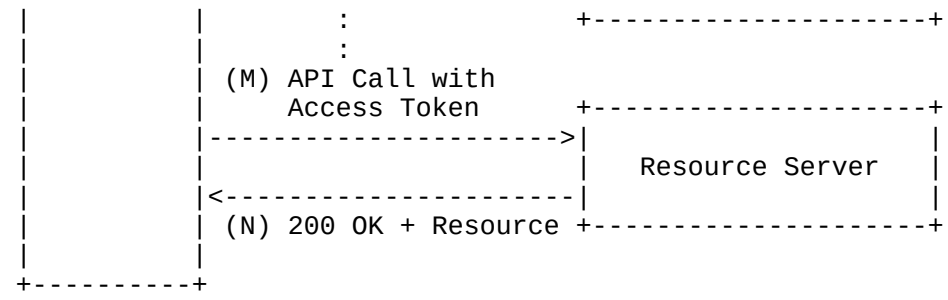


Figure: Client remediates using metadata of required authorization details types

* (A) The user starts the flow.

* (B) The client calls an API with an access token.

* (C) Resource server returns HTTP 403 Forbidden including a WWW-Authenticate header with error code insufficient_authorization_details and the resource metadata url (OAuth 2.0 Protected Resource Metadata [RFC9728]).

* (D-E) The client discovers expected authorization details types from resource metadata endpoint’s response.

* (F-G) The client consumes authorization details types metadata from authorization server’s authorization_details_types_metadata_endpoint.

* (H-I) The client constructs a valid authorization details object and makes an OAuth + RAR [RFC9396] request.

* (J) Authorization server returns authorization code.

* (K-L) The client exchanges authorization code for access token.

* (M) The client makes an API request with the (RAR) access token.

* (N) Resource server validates access token and returns successful response.

3.2. Client remediates using actionable authorization details objects provided by resource server

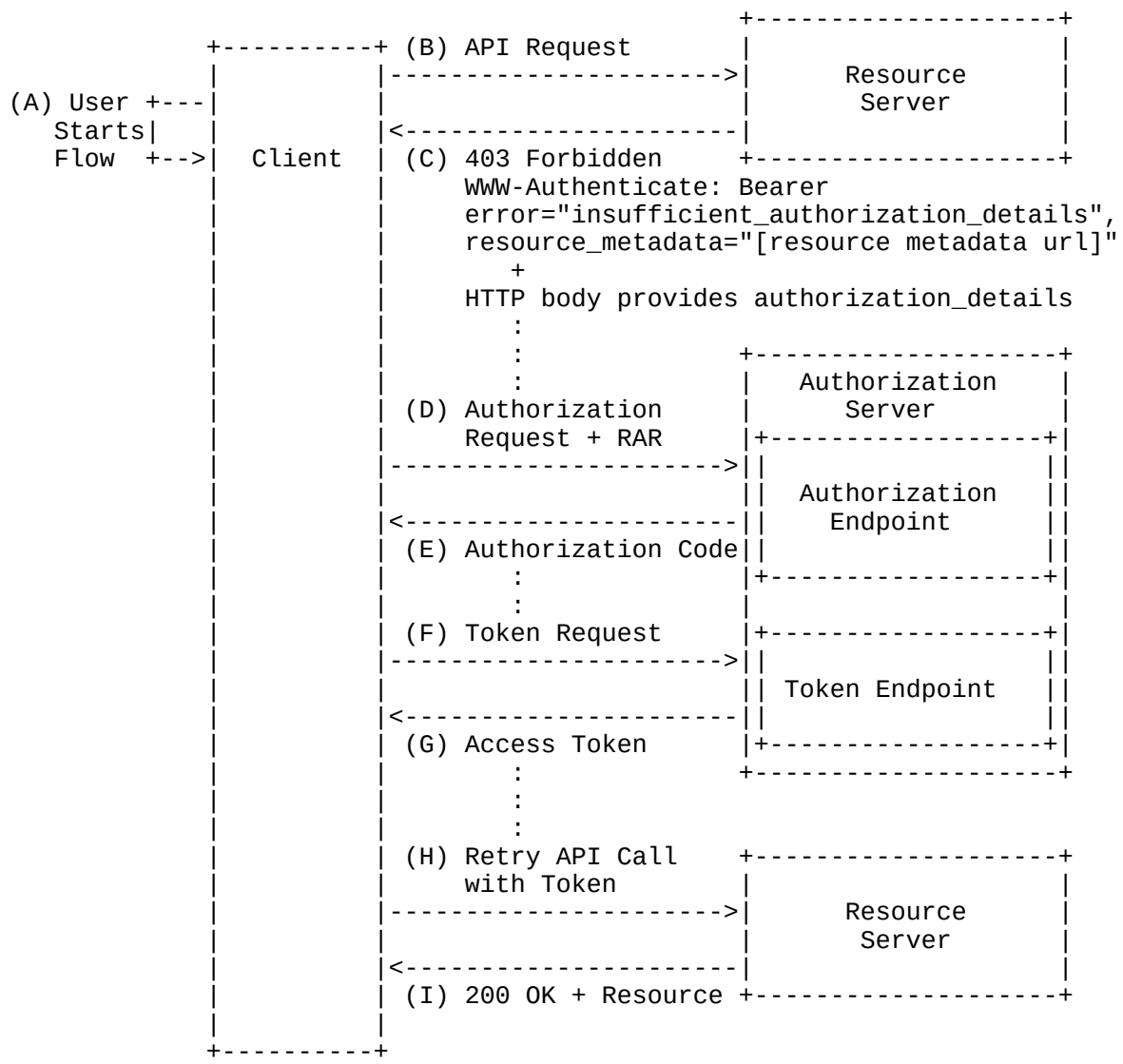


Figure: Client remediates using actionable authorization details objects provided by resource server

* (A) The user starts the flow.

* (B) The client calls an API with an access token.

* (C) Resource server returns HTTP 403 Forbidden including a WWW-Authenticate header with error code insufficient_authorization_details and in the response body includes the *required authorization details objects*.

* (D) The client uses the obtained authorization details objects in a new OAuth + RAR [RFC9396] request.

* (E) Authorization server returns authorization code.

* (F-G) The client exchanges authorization code for access token.

* (H) The client makes an API request with the (RAR) access token.

* (I) Resource server validates access token and returns successful response.

4. OAuth 2.0 Protected Resource Metadata [RFC9728]

This document specifies a new OPTIONAL metadata attribute: authorization_details_types_required, to be included in the response of OAuth Protected Resource Metadata [RFC9728].

"authorization_details_types_required": RECOMMENDED. a JSON object that conforms to the syntax described in Section 4.1 for a _required types expression_.

The following is a non-normative example response with the new authorization_details_types_required attribute:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "resource":
    "https://resource.example.com/payments",
    "authorization_servers":
        ["https://as1.example.com",
        "https://as2.example.net"],
    "bearer_methods_supported": ["header"],
    "scopes_supported": ["payment"],
    "resource_documentation":
        "https://resource.example.com/docs/payments.html",
    "authorization_details_types_required": {
        "oneOf": ["payment_initiation", "payment_approval",
                  "beneficiary_designation"]
    }
}
```

Note: When resource servers accept access tokens _from several authorization servers_, clients can discover which authorization details types each authorization server supports.

4.1. Required types expression syntax

The following JSON syntax defines a *required types expression* that describes permitted combinations of required _authorization_details_ types. This expression allows selection operators (oneOf, allOf) and boolean composition (and, or) to be combined in a predictable manner.

A *required types expression* is a JSON object whose top-level claims MUST contain *exactly* one of the following attributes:

* and
* or
* oneOf
* allOf

Attributes definition:

"and": OPTIONAL. a non-empty JSON array of _required types expressions_. When *and* is specified, the expression is satisfied if *all* contained expressions are satisfied.

"or": OPTIONAL. a non-empty JSON array of _required types expressions_. When *or* is specified, the expression is satisfied if *at least one* contained expression is satisfied.

"oneOf": OPTIONAL. a non-empty JSON array of strings identifying authorization_details types. When *oneOf* is specified, the expression is satisfied if *exactly one* of the listed types is present.

"allOf": OPTIONAL. a non-empty JSON array of strings identifying authorization_details types. When *allOf* is specified, the expression is satisfied if *all* of the listed types are present.

4.2. Required types expression examples

4.2.1. Example expression using "and" operator

Specifies that the selection MUST include a and b, *and* one of c *or* d.

```
{
  "and": [
    { "allOf": ["a", "b"] },
    { "oneOf": ["c", "d"] }
  ]
}
```

4.2.2. Example expression using "or" operator

Specifies that the selection MUST include *either* c *and* d, *or* one of a or b.

```
{
  "or": [
    { "allOf": ["c", "d"] },
    { "oneOf": ["a", "b"] }
  ]
}
```

5. Authorization Details Types Metadata Endpoint

The following authorization server metadata [RFC8414] parameter is introduced to signal the server’s support for Authorization Details Types Metadata:

"authorization_details_types_metadata_endpoint": OPTIONAL. The URL of the Authorization Details Types Metadata endpoint.

5.1. Authorization Details Types Metadata Endpoint Response

The Authorization Details Types Metadata endpoint’s response is a JSON document with the key authorization_details_types_metadata whose attributes are authorization details type identifiers.

Each identifier is an object describing a single authorization details type.

```
{
  "authorization_details_types_metadata": {
    "type": {
      "version": "...",
      "description": "...",
      "documentation_uri": "...",
      "schema": { },
      "schema_uri": "...",
      "examples": [ ]
    }
  }
}
```

Attributes definition:

"version": OPTIONAL. String identifying the version of the authorization details type definition. The value is informational and does not imply semantic version negotiation.

"description": OPTIONAL. String containing a description of the authorization details type. Clients MUST NOT rely on this value for authorization or validation decisions.

"documentation_uri": OPTIONAL. URI referencing external documentation describing the authorization details type.

"schema": The schema attribute contains a JSON Schema document [JSON.Schema] that describes a single authorization details object. The schema MUST validate exactly one authorization details object and MUST restrict the type attribute to the corresponding authorization details type identifier. This attribute is REQUIRED unless schema_uri is specified. If present, schema_uri MUST NOT be included.

"schema_uri": The schema_uri attribute is an absolute URI, as defined by RFC 3986 [RFC3986], referencing a JSON Schema document describing a single authorization details object. The referenced schema MUST satisfy the same requirements as the schema attribute. This attribute is REQUIRED unless schema is specified. If this attribute is present, schema MUST NOT be present.

"examples": OPTIONAL. An array of example authorization details objects. Examples are non-normative.

See Examples Appendix A.1 for non-normative response example.

6. Resource Server Error Signaling of insufficient authorization_details

This document defines a new error code in the OAuth 2.0 WWW-Authenticate Error Code Registry, insufficient_authorization_details, which resource servers SHALL return using the WWW-Authenticate header, to signal access is denied due to missing or insufficient authorization details.

Example HTTP response:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_authorization_details",
    resource_metadata="https://resource.example.com/
    .well-known/oauth-protected-resource/payments"
```

6.1. OPTIONAL authorization_details in response body

Resource server MAY include, alongside the insufficient_authorization_details error, an informative HTTP response body with content type application/json containing the required authorization details objects needed to satisfy the failing request.

Note:

* The audience of authorization details objects provided by a resource server in an error response is its trusted authorization servers, as advertised by the Resource Server’ s metadata endpoint.

* Resource servers SHOULD provide authorization_details objects only if *all* trusted authorization servers accept the *authorization details type* used.

HTTP response body definition:

"authorization_details": OPTIONAL. Array of authorization details objects, matching the format specified in RAR [RFC9396] for the authorization_details request parameter.

"authorization_hint": RECOMMENDED. String serving as a stable reference, enabling the client to select existing access tokens linked to authorization details objects without having to understand RAR object semantics. Its value SHALL be identical for semantically equal authorization_details and it SHALL NOT be returned in case tokens resulting from provided authorization_details are single-use only.

Clients MAY use the provided authorization_details in a subsequent OAuth request to obtain an access token satisfying the resource's requirements.

Example resource server response with OPTIONAL authorization_details:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_authorization_details",
    resource_metadata="https://resource.example.com/
    .well-known/oauth-protected-resource/payments"
Content-Type: application/json
Cache-Control: no-store

{
  "authorization_details": [{
    "type": "payment_initiation",
    "instructed_amount": {
      "currency": "EUR",
      "amount": "100.00"
    },
    "creditor_account": {
      "iban": "DE02120300000000202051"
    }
  }],
  "authorization_hint": "Yb7q3AC5d"
}
```

7. Handling large RAR objects when issuing access tokens

RAR [RFC9396] section 9 instructs that authorization servers MUST provide approved RAR objects to resource servers for enforcement. The authorization server MAY add the authorization_details attribute to access tokens in JSON Web Token (JWT) format or to token introspection responses.

Including large RAR objects in JWT access tokens may cause interoperability loss due to token sizes exceeding header size restrictions.

Authorization servers SHOULD support a configurable *maximum approved RAR objects size threshold* (in bytes). If the size exceeds this threshold, JWT access tokens SHALL NOT include the authorization_details claim; instead, approved authorization details will be accessed via token introspection [RFC7662].

8. Processing Rules

8.1. Client Processing Rules

* When receiving error insufficient_authorization_details, if response body contains an _authorization_hint_ claim that matches a valid token in client's possession, client SHOULD retry calling the failing endpoint using the matching token.

* If the response body contains an _authorization_details_ claim, the client MAY include it in a subsequent OAuth request to obtain a token, which it MAY then use to retry the failing endpoint.

* Otherwise, the client MAY consult metadata:

  - Fetch resource metadata to discover accepted authorization servers and required *authorization_details types*.

  - Fetch authorization server metadata to discover authorization_details_types_supported.

  - Fetch authorization server's authorization_details_types_metadata_endpoint to obtain authorization details type metadata and schemas.

  - Locate schema or retrieve schema_uri.

* Construct authorization details conforming to the schema and include in subsequent OAuth request to obtain a token with which it MAY retry calling the failing endpoint.

8.2. Resource Server Processing Rules

* Advertise in resource metadata authorization_details_types_required, where relevant.

* Verify access tokens against required authorization details.

* If insufficient, the resource server MUST return HTTP 403 with WWW-Authenticate: Bearer error="insufficient_authorization_details".

* OPTIONALLY provide also an HTTP body with an informative actionable authorization_details object.

9. Security Considerations

9.1. Cacheability and Intermediaries

HTTP 403 responses with response bodies may be cached or replayed in unexpected contexts. Recommended mitigation is resource servers SHALL use Cache-Control: no-store response header.

9.2. Confidentiality of resource server provided authorization_details

Resource server providing actionable authorization_details SHOULD NOT include sensitive data within them. This is consistent with RAR [RFC9396] authorization_details OAuth request parameter, representing *request* semantics.

Confidentiality-preserving authorization_details types SHOULD NOT include sensitive data. Instead, end-user SHALL provide such information when interacting with the authorization server.

Alternatively, authorization_details MAY refer to specific end-user resources using opaque reference handles (e.g "account_1a" instead of using explicit IBAN).

10. IANA Considerations

10.1. OAuth 2.0 WWW-Authenticate Error Code Registry

Error Code	Description
insufficient_authorization_details	The request is missing required authorization details or the provided authorization details are insufficient.

Table 1

10.2. OAuth Metadata Attribute Registration

The metadata attribute authorization_details_types_metadata_endpoint is defined for OAuth 2.0 authorization server metadata as a URL. The metadata attribute authorization_details_types_required is defined for OAuth 2.0 protected resource metadata [RFC9728].

11. Normative References

[IANA.oauth-parameters]
IANA, "OAuth Parameters",
<https://www.iana.org/assignments/oauth-parameters>.

[JSON.Schema]
Wright, Ed, A., Andrews, Ed, H., Hutton, Ed Postman, B.,
and G. Dennis, "JSON Schema: A Media Type for Describing
JSON Documents", June 2022,
<https://json-schema.org/draft/2020-12/json-schema-core>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
Resource Identifier (URI): Generic Syntax", STD 66,
RFC 3986, DOI 10.17487/RFC3986, January 2005,
<https://www.rfc-editor.org/rfc/rfc3986>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework",
RFC 6749, DOI 10.17487/RFC6749, October 2012,
<https://www.rfc-editor.org/rfc/rfc6749>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection",
RFC 7662, DOI 10.17487/RFC7662, October 2015,
<https://www.rfc-editor.org/rfc/rfc7662>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8414] Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0
Authorization Server Metadata", RFC 8414,
DOI 10.17487/RFC8414, June 2018,
<https://www.rfc-editor.org/rfc/rfc8414>.

[RFC9396] Lodderstedt, T., Richer, J., and B. Campbell, "OAuth 2.0
Rich Authorization Requests", RFC 9396,
DOI 10.17487/RFC9396, May 2023,
<https://www.rfc-editor.org/rfc/rfc9396>.

[RFC9728] Jones, M.B., Hunt, P., and A. Parecki, "OAuth 2.0
Protected Resource Metadata", RFC 9728,
DOI 10.17487/RFC9728, April 2025,
<https://www.rfc-editor.org/rfc/rfc9728>.

Appendix A. Examples

This section provides non-normative examples of how this specification may be used to support specific use cases.

A.1. Authorization Server Metadata Examples

A.1.1. Example authorization_details_types_metadata_endpoint response with Payment Initiation

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "authorization_details_types_metadata": {
        "payment_initiation": {
            "version": "1.0",
            "description": "Authorization to initiate a single payment from a payer accou
nt to a creditor account.",
            "documentation_uri": "https://example.com/docs/payment-initiation",
            "schema": {
                "$schema": "https://json-schema.org/draft/2020-12/schema",
                "title": "Payment Initiation Authorization Detail",
                "type": "object",
                "required": [
                    "type",
                    "instructed_amount",
                    "creditor_account"
                ],
                "properties": {
                    "type": {
                        "const": "payment_initiation",
                        "description": "Authorization details type identifier."
                    },
                    "actions": {
                        "type": "array",
                        "description": "Permitted actions for this authorization.",
                        "items": {
                            "type": "string",
                            "enum": ["initiate"]
                        },
                        "minItems": 1,
                        "uniqueItems": true
                    },
                    "instructed_amount": {
                        "type": "object",
                        "description": "Amount and currency of the payment to be initiate
d.",
                        "required": ["currency", "amount"],
                        "properties": {
```

```
                            "currency": {
                                "type": "string",
                                "description": "ISO 4217 currency code.",
                                "pattern": "^[A-Z]{3}$"
                            },
                            "amount": {
                                "type": "string",
                                "description": "Decimal monetary amount represented as a
string.",
                                "pattern": "^[0-9]+(\\.[0-9]{1,2})?$"
                            }
                        },
                        "additionalProperties": false
                    },
                    "creditor_account": {
                        "type": "object",
                        "description": "Account to which the payment will be credited.",
                        "required": ["iban"],
                        "properties": {
                            "iban": {
                                "type": "string",
                                "description": "International Bank Account Number (IBAN).
",
                                "pattern": "^[A-Z0-9]{15,34}$"
                            }
                        },
                        "additionalProperties": false
                    },
                    "remittance_information": {
                        "type": "string",
                        "description": "Unstructured remittance information for the payme
nt.",
                        "maxLength": 140
                    }
                },
                "additionalProperties": false
            }
        }
    }
}
```

A.1.2. Example authorization_details_types_metadata_endpoint response for the Norwegian Health Sector (HelseID)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "authorization_details_types_metadata": {
        "helseid_authorization": {
            "version": "1.0",
```

```
            "description": "Allows the OAuth client to pass organization information to H
elseID.",
            "documentation_uri": "https://utviklerportal.nhn.no/informasjonstjenester/hel
seid/bruksmoenstre-og-eksempelkode/bruk-av-helseid/docs/tekniske-mekanismer/organisasjons
numre_enmd",
            "schema": {
                "$schema": "http://json-schema.org/draft-07/schema#",
                "title": "Organization numbers for a multi-tenant client",
                "type": "object",
                "properties": {
                    "type": {
                        "type": "string",
                        "const": "helseid_autorization"
                    },
                    "practitioner_role": {
                        "type": "object",
                        "properties": {
                            "organization": {
                                "type": "object",
                                "properties": {
                                    "identifier": {
                                        "type": "object",
                                        "properties": {
                                            "system": {
                                                "type": "string"
                                            },
                                            "type": {
                                                "type": "string"
                                            },
                                            "value": {
                                                "type": "string"
                                            }
                                        },
                                        "required": [
                                            "system",
                                            "type",
                                            "value"
                                        ]
                                    }
                                },
                                "required": [
                                    "identifier"
                                ]
                            }
                        },
                        "required": [
                            "organization"
                        ]
                    }
                },
                "required": [
```

```
                    "type",
                    "practitioner_role"
                ]
            }
        },
        "helseid_trust_framework": {
            "$schema": "http://json-schema.org/draft-07/schema#",
            "description": "Complete Trust Framework structure",
            "documentation_uri": "https://utviklerportal.nhn.no/informasjonstjenester/hel
seid/bruksmoenstre-og-eksempelkode/bruk-av-helseid/docs/tillitsrammeverk/profil_for_tilli
tsrammeverkmd",
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "const": "nhn:tillitsrammeverk:parameters"
                },
                "practitioner": {
                    "type": "object",
                    "properties": {
                        "authorization": {
                            "type": "object",
                            "properties": {
                                "code": {
                                    "type": "string"
                                },
                                "system": {
                                    "type": "string"
                                }
                            },
                            "required": [
                                "code",
                                "system"
                            ]
                        },
                        "legal_entity": {
                            "type": "object",
                            "properties": {
                                "id": {
                                    "type": "string"
                                },
                                "system": {
                                    "type": "string"
                                }
                            },
                            "required": [
                                "id",
                                "system"
                            ]
                        },
```

```
                    "point_of_care": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "string"
                            },
                            "system": {
                                "type": "string"
                            }
                        },
                        "required": [
                            "id",
                            "system"
                        ]
                    },
                    "department": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "string"
                            },
                            "system": {
                                "type": "string"
                            }
                        },
                        "required": [
                            "id",
                            "system"
                        ]
                    }
                },
                "required": [
                    "authorization",
                    "legal_entity",
                    "point_of_care",
                    "department"
                ]
            },
            "care_relationship": {
                "type": "object",
                "properties": {
                    "healthcare_service": {
                        "type": "object",
                        "properties": {
                            "code": {
                                "type": "string"
                            },
                            "system": {
```

```
                    "type": "string"
                }
            },
            "required": [
                "code",
                "system"
            ]
        },
        "purpose_of_use": {
            "type": "object",
            "properties": {
                "code": {
                    "type": "string"
                },
                "system": {
                    "type": "string"
                }
            },
            "required": [
                "code",
                "system"
            ]
        },
        "purpose_of_use_details": {
            "type": "object",
            "properties": {
                "code": {
                    "type": "string"
                },
                "system": {
                    "type": "string"
                }
            },
            "required": [
                "code",
                "system"
            ]
        },
        "decision_ref": {
            "type": "object",
            "properties": {
                "id": {
                    "type": "string"
                },
                "user_selected": {
                    "type": "boolean"
                }
            },
```

```
                    "required": [
                        "id",
                        "user_selected"
                    ]
                }
            },
            "required": [
                "healthcare_service",
                "purpose_of_use",
                "purpose_of_use_details",
                "decision_ref"
            ]
        },
        "patients": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "point_of_care": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "string"
                            },
                            "system": {
                                "type": "string"
                            }
                        },
                        "required": [
                            "id",
                            "system"
                        ]
                    },
                    "department": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "string"
                            },
                            "system": {
                                "type": "string"
                            }
                        },
                        "required": [
                            "id",
                            "system"
                        ]
                    }
```

```
                        },
                        "required": [
                            "point_of_care",
                            "department"
                        ]
                    }
                }
            },
            "required": [
                "type",
                "practitioner",
                "care_relationship",
                "patients"
            ]
        }
    }
}
```

A.2. Protected Resource Metadata Examples

A.2.1. Example Protected Resource Metadata response of payments resource

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "resource": "https://resource.example.com/payments",
    "authorization_servers":
        ["https://as1.example.com",
        "https://as2.example.net"],
    "bearer_methods_supported": ["header"],
    "scopes_supported": ["payment"],
    "resource_documentation":
        "https://resource.example.com/docs/payments.html",
    "authorization_details_types_required": {
        "oneOf": ["payment_initiation", "payment_approval",
                  "beneficiary_designation"]
    }
}
```

A.2.2. Example Protected Resource Metadata response from the Norwegian Health Sector (HelseID)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "resource": "https://health-api.nhn.no/health-information",
    "authorization_servers": ["https://helseid-sts.nhn.no"],
    "bearer_methods_supported": ["header"],
    "scopes_supported":
        ["nhn:health-api/read", "nhn:health-api/write"],
    "resource_documentation": "https://utviklerportal.nhn.no",
    "authorization_details_types_required": {
        "allOf": ["helseid_authorization",
                  "nhn:tillitsrammeverk:parameters"]
    }
}
```

A.3. Payment initiation with RAR error signaling

A.3.1. Client initiates API request

Client uses access token obtained at login to call payment initiation API

```
POST /payments HTTP/1.1
Host: resource.example.com
Content-Type: application/json
Authorization: Bearer eyj... (access token from login)

{
    "type": "payment_initiation",
    "locations": [
        "https://resource.example.com/payments"
    ],
    "instructed_amount": {
        "currency": "EUR",
        "amount": "123.50"
    },
    "creditor_name": "Merchant A",
    "creditor_account": {
        "bic": "ABCIDEFFXXX",
        "iban": "DE02100100109307118603"
    }
}
```

A.3.2. Resource server signals insufficient_authorization_details with actionable RAR object

Resource server requires payment approval and responds with:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_authorization_details",
    resource_metadata="https://resource.example.com
    /.well-known/oauth-protected-resource/payments"
Content-Type: application/json
Cache-Control: no-store

{
    "authorization_details": [{
      "type": "payment_initiation",
      "locations": [
          "https://example.com/payments"
      ],
      "instructed_amount": {
          "currency": "EUR",
          "amount": "123.50"
      },
      "creditor_name": "Merchant A",
      "creditor_account": {
          "bic": "ABCIDEFFXXX",
          "iban": "DE02100100109307118603"
      },
      "interaction_id": "f81d4fae-7dec-11d0-a765-00a0c91e6bf6",
      "risk_profile": "B-71"
}]
}
```

Note: the resource server has added the ephemeral attributes interaction_id and risk_profile.

A.3.3. Client initiates OAuth flow using the provided authorization_details object

After user approves the request, client obtains an access token representing the approved payment

A.3.4. Client re-attempts API request

```
POST /payments HTTP/1.1
Host: resource.example.com
Content-Type: application/json
Authorization: Bearer eyj... (payment approval access token)

{
    "type": "payment_initiation",
    "locations": [
        "https://resource.example.com/payments"
    ],
    "instructed_amount": {
        "currency": "EUR",
        "amount": "123.50"
    },
    "creditor_name": "Merchant A",
    "creditor_account": {
        "bic": "ABCIDEFFXXX",
        "iban": "DE02100100109307118603"
    }
}
```

A.3.5. Resource server authorizes the request

```
HTTP/1.1 201 Accepted
Content-Type: application/json
Cache-Control: no-store

{
    "paymentId": "a81bc81b-dead-4e5d-abff-90865d1e13b1",
    "status": "accepted"
}
```

Appendix B. Document History

-03

* Added authorization_hint to guide client on token selection and updated client processing rules accordingly

* Added security consideration on confidentiality of RS-provided authorization_details

* Added authorization server considerations for handling large RAR objects in JWT access tokens

-02

* Defined the required types expression

* Added Protected Resource Metadata examples

-01

* Authorization details moved to HTTP body and made OPTIONAL

* Metadata pointer from resource metadata url, full authorization details types metadata on authorization server new endpoint

-00

* Document creation

Acknowledgments

The authors would like to thank the following individuals who contributed ideas, feedback, and wording that helped shape the final specification: Rune Grimstad.

Author’s Address

Yaron Zehavi
Raiffeisen Bank International
Email: yaron.zehavi@rbinternational.com